

# Pemanfaatan Neural Network Perceptron Pada Pengenalan Pola Karakter

Kukuh Yudhistiro  
kukuh.yudhistiro@unmer.ac.id

Fakultas Teknologi Informasi Universitas Merdeka Malang

**Abstract**— Various methods on artificial neural network has been applied to identify patterns of characters one using Perceptron algorithm. Perceptron algorithm can be used to identify patterns of characters with various input patterns that resemble letters of the alphabet so perceptron can be trained to recognize it. Perceptron algorithm is an algorithm in a neural network that includes supervised neural network.

**Intisari**— Berbagai metode pada jaringan syaraf tiruan telah diterapkan dengan menggunakan algoritma Perceptron. Algoritma Perceptron dapat digunakan untuk mengidentifikasi pola karakter yang dapat dilatih perceptron untuk mengenalinya. Algoritma perceptron adalah algoritma dalam jaringan syaraf tiruan yang mencakup jaringan syaraf tiruan.

**Kata Kunci**— neural network, perceptron, pengenalan pola, algoritma perceptron, deteksi fitur

## I. PENDAHULUAN

*Character recognition* merupakan bidang yang termasuk dalam penelitian berbasis jaringan syaraf tiruan. Pada algoritma perceptron vektor pola yang diinputkan adalah vektor pola karakter A, B dan C yang bervariasi. Dari pola yang sudah diberikan, maka langkah selanjutnya adalah melatih jaringan tersebut agar memperoleh bobot ( $w$ ) dan bias ( $b$ ) yang diinginkan sehingga dapat membandingkan keluaran jaringan yang didapat dengan target yang sudah ditentukan. Hasil percobaan penulis didapatkan bahwa pengenalan pola A, B, C dapat dikenali cukup dalam 3 epoch dengan nilai bobot dan nilai bias tertentu [1][4].

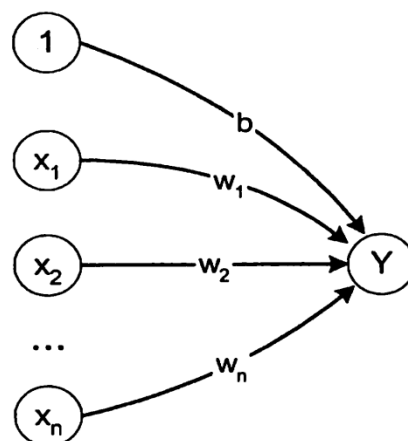
## II. KAJIAN PUSTAKA

### A. Algoritma Perceptron

Model jaringan perceptron ditemukan oleh Rosenblatt (1962) dan Minsky – Papert (1969). Model tersebut merupakan model yang memiliki aplikasi dan pelatihan yang paling baik pada era tersebut [2].

Arsitektur jaringan perceptron mirip dengan arsitektur jaringan Hebb disajikan pada gambar 1.

Jaringan terdiri dari beberapa unit masukan dan memiliki sebuah unit keluaran. Hanya saja fungsi aktivasi bukan merupakan fungsi biner (atau bipolar), tetapi memiliki kemungkinan nilai -1,0 atau 1.



**Gambar 1.** Diagram jaringan perceptron

Untuk suatu harga *threshold*  $\theta$  yang ditentukan:

$$f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > \theta \\ 0 & \text{jika } -\theta \leq \text{net} \leq \theta \\ -1 & \text{jika } \text{net} < -\theta \end{cases} \quad (1)$$

Secara geometris, fungsi aktivasi membentuk 2 garis sekaligus masing-masing dengan persamaan:

$$\begin{aligned} w_1x_1 + w_2x_2 + \dots + w_nx_n + b &= \theta \quad \text{dan} \\ w_1x_1 + w_2x_2 + \dots + w_nx_n + b &= -\theta \end{aligned} \quad (2)$$

### B. Training Pada Algoritma Perceptron

Misalkan:

- $s$  adalah vektor inputan
- $t$  adalah target output
- $\alpha$  adalah *learning rate* (laju pemahaman yang telah ditentukan)
- $\theta$  adalah *threshold* yang ditentukan

maka algoritma pelatihan perceptron adalah:

1. inisialisasi semua bobot dan bias ( $=\alpha$ ). Biasanya *learning rate* diberi nilai = 1
2. selama ada elemen vektor masukan yang respon unit keluaran tidak sama dengan target, lakukan:

- a. set aktivasi unit masukan:
 
$$x_i = s_i \quad (i = 1, \dots, n) \quad (3)$$

- b. hitung respon unit keluaran:
 
$$\text{net} = \sum_i x_i w_i + b \quad (4)$$

$$y = f(\text{net}) = \begin{cases} 1 & \text{jika } \text{net} > \theta \\ 0 & \text{jika } -\theta \leq \text{net} \leq \theta \\ -1 & \text{jika } \text{net} < -\theta \end{cases} \quad (5)$$

- c. Revisi bobot pola yang mengandung error ( $y \neq t$ ) dengan persamaan:

$$w_i(\text{baru}) = w_i(\text{lama}) + \Delta w \quad (i=1, \dots, n) \text{ dengan } \Delta w = \alpha t x_i$$

$$b(\text{baru}) = b(\text{lama}) + \Delta b \text{ dengan } \Delta b = \alpha t \quad (6)$$

### C. Proses Pelatihan

1. Iterasi dilakukan terus sampai semua pola memiliki keluaran jaringan yang sama dengan targetnya (jaringan sudah memahami pola). Iterasi tidak berhenti setelah semua pola dimasukkan.
2. Perubahan bobot hanya dilakukan pada pola yang mengandung kesalahan. Perubahan tersebut merupakan hasil kali unit masukan dengan target dan *learning rate*. Perubahan bobot hanya akan terjadi kalau unit masukan  $\neq 0$ .
3. Kecepatan iterasi ditentukan oleh *learning rate* ( $0 \leq \alpha \leq 1$ ) yang dipakai. Semakin besar harga *learning rate* semakin sedikit iterasi yang diperlukan. Akan tetapi jika *learning rate* terlalu besar maka akan merusak pola yang sudah benar sehingga laju pemahaman menjadi lambat.

Algoritma pelatihan perceptron lebih baik dibandingkan model Hebb [1][3] karena:

1. setiap kali sebuah pola dimasukkan, hasil keluaran jaringan dibandingkan dengan target yang sesungguhnya. Jika terdapat perbedaan, maka bobot akan dimodifikasi. Jadi tidak semua bobot selalu dimodifikasi dalam setiap iterasinya.
2. Modifikasi bobot tidak hanya ditentukan oleh perkalian antara target dengan masukan, tetapi juga melibatkan suatu laju pemahaman (*learning rate*) yang nilainya dapat diatur.
3. Pelatihan dilakukan berulang-ulang untuk semua kemungkinan pola yang ada hingga jaringan dapat mengerti polanya. Satu siklus pelatihan yang melibatkan semua pola disebut *epoch*.

### III. METODE PENELITIAN

Algoritma untuk mengenali apakah pola inputan yang diberikan menyerupai sebuah karakter A, B atau C atau tidak adalah:

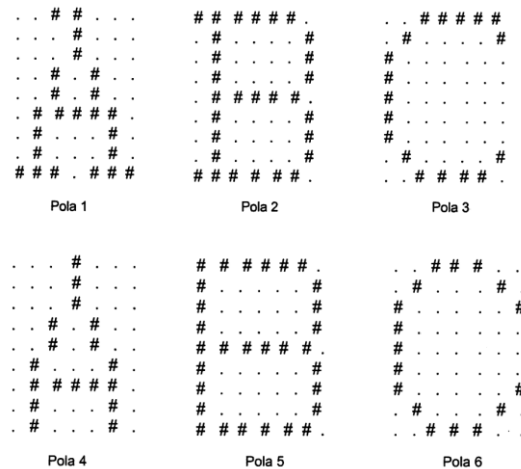
1. Nyatakan tiap pola inputan sebagai matrik bipolar yang elemennya adalah tiap titik dalam pola tersebut
2. berikan nilai target  $=+1$  jika pola inputan menyerupai huruf yang diinginkan. Jika sebaliknya, berikan nilai target  $= -1$
3. Berikan inisialisasi bobot, bias, laju pemahaman (*learning rate*) dan *threshold*.
4. Lakukan proses pelatihan.

### IV. HASIL DAN PEMBAHASAN

Untuk menentukan vektor masukan, tiap titik dalam pola diambil sebagai komponen vektor. Jadi setiap vektor masukan memiliki  $9 \times 7 = 63$  komponen. Titik dalam pola yang bertanda “#” diberi nilai  $= +1$  dan yang bertanda “.” Diberi nilai  $= -1$ .

Pembacaan pola dilakukan dari kiri ke kanan, dimulai dari baris paling atas.

Berikut adalah pola-pola inputan yang diberikan:



Gambar 2. Pola yang akan diuji

Vektor masukan pola 1:

'	-1	-1	1	1	-1	-1	-1
'	-1	-1	-1	1	-1	-1	-1
'	-1	-1	-1	1	-1	-1	-1
'	-1	-1	1	-1	1	-1	-1
'	-1	-1	1	-1	1	-1	-1
'	-1	1	1	1	1	1	-1
'	-1	1	-1	-1	-1	1	-1
'	-1	1	-1	-1	-1	1	-1
'	1	1	1	-1	1	1	1

Gambar 3. Pola ke-1

Vektor masukan pola 2:

'	1	1	1	1	1	1	-1
'	-1	1	-1	-1	-1	-1	1
'	-1	1	-1	-1	-1	-1	1
'	-1	1	-1	-1	-1	-1	1
'	-1	1	1	1	1	1	-1
'	-1	1	-1	-1	-1	-1	1
'	-1	1	-1	-1	-1	-1	1
'	-1	1	-1	-1	-1	-1	1
'	1	1	1	1	1	1	-1

Gambar 4. Pola ke-2

Vektor masukan pola 3:

'	-1	-1	1	1	1	1	1
'	-1	1	-1	-1	-1	-1	1
'	1	-1	-1	-1	-1	-1	-1
'	1	-1	-1	-1	-1	-1	-1
'	1	-1	-1	-1	-1	-1	-1
'	1	-1	-1	-1	-1	-1	-1
'	1	-1	-1	-1	-1	-1	-1
'	1	-1	-1	-1	-1	-1	-1
'	-1	1	-1	-1	-1	-1	1
'	-1	-1	1	1	1	1	-1

Gambar 5. Pola ke-3



Gambar 12. Epoch ke 1 (iterasi ke 0)

Selama  $f(\text{net}) < t$  atau  $y < t$ , iterasi dilanjutkan. User dapat melakukan perubahan parameter:

```
C:\Windows\system32\cmd.exe
*****
Implementasi Algoritma Perceptron
Pada Pengenalan Pola
Kukuh Yudhistiro
*****
Masukkan Learning Rate      : 1
Masukkan Threshold          : 0.2
Masukkan Maks Iterasi       : 6
bias = 0
*****
```

Gambar 13. Pemberian nilai kembali untuk uji ulang

Masukan ( $x_1 \ x_2 \ 1$ )	Target $t$	$y$ $= f(\text{net})$	Perubahan Bobot ( $\Delta w_1 \ \Delta w_2 \ \Delta b$ )	Bobot Baru ( $w_1 \ w_2 \ b$ )
inisialisasi				(0 0 0)
(1 1 1)	1	0	(1 1 1)	(1 1 1)
(1 0 1)	-1	2	(-1 0 -1)	(0 1 0)
(0 1 1)	-1	1	(0 -1 -1)	(0 0 -1)
(0 0 1)	-1	-1	(0 0 0)	(0 0 -1)

Gambar 14. Perhitungan secara tabel

Iterasi ke = 0 Pola ke: 0 net = 0 Bobot berubah = 1 Pola ke: 1 net = 1 Bobot berubah = 0 Pola ke: 2 net = 2 Bobot berubah = -1 Pola ke: 3 net = -3 Bobot tidak berubah = -1	Iterasi ke = 1 Pola ke: 0 net = 1 Bobot berubah = 0 Pola ke: 1 net = 0 Bobot berubah = -1 Pola ke: 2 net = 1 Bobot berubah = -2 Pola ke: 3 net = -6 Bobot tidak berubah = -2
---	--

Gambar 15. Epoch 1 dan 2

Iterasi ke = 2 Pola ke: 0 net = 2 Bobot berubah = -1 Pola ke: 1 net = -1 Bobot berubah = -2 Pola ke: 2 net = 0 Bobot berubah = -3 Pola ke: 3 net = -9 Bobot tidak berubah = -3	Iterasi ke = 3 Pola ke: 0 net = 3 Bobot tidak berubah = -3 Pola ke: 1 net = -3 Bobot tidak berubah = -3 Pola ke: 2 net = -3 Bobot tidak berubah = -3 Pola ke: 3 net = -9 Bobot tidak berubah = -3
--	---

Gambar 16. Epoch 3 dan 4

Iterasi ke = 9 Pola ke: 0 net = 3 Bobot tidak berubah = -3 Pola ke: 1 net = -3 Bobot tidak berubah = -3 Pola ke: 2 net = -3 Bobot tidak berubah = -3 Pola ke: 3 net = -9 Bobot tidak berubah = -3 selesai
--

Gambar 17. Epoch 10

Dari gambar 17 di atas, bobot tidak berubah lagi sejak epoch 4 (iterasi 3) sampai epoch 10 (iterasi 9)

```
Mapping y=f(net)=1 jika net>2
y=0 jika -2<net<2 ; y=-1 jika net<-2

Pola ke = 1
net = 56 maka y = 1

Pola ke = 2
net = -56 maka y = -1

Pola ke = 3
net = -12 maka y = -1

Pola ke = 4
net = 40 maka y = 1

Pola ke = 5
net = -44 maka y = -1

Pola ke = 6
net = -4 maka y = -1

Target Pola 1 = 1
Target Pola 2 = -1
Target Pola 3 = -1
Target Pola 4 = 1
Target Pola 5 = -1
Target Pola 6 = -1
```

Gambar 18. Hasil perhitungan net dibandingkan dengan Target pola

Dari gambar di atas menunjukkan target pola yang diharapkan sama dengan target awal:

Pola Masukan	Target
Pola 1	1
Pola 2	-1
Pola 3	-1
Pola 4	1
Pola 5	-1
Pola 6	-1

Gambar 19. Hasil perhitungan net dibandingkan dengan Target pola

```
BENTUK HASIL INPUT POLA 1 DIUBAH JADI ARRAY 7:9 =
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .

BENTUK HASIL INPUT POLA 2 DIUBAH JADI ARRAY 7:9 =
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .

BENTUK HASIL INPUT POLA 3 DIUBAH JADI ARRAY 7:9 =
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
```

Gambar 20. Tampil hasil pengenalan karakter pola 1,2 dan 3

```
BENTUK HASIL INPUT POLA 4 DIUBAH JADI ARRAY 7:9 =
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .

BENTUK HASIL INPUT POLA 5 DIUBAH JADI ARRAY 7:9 =
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .

BENTUK HASIL INPUT POLA 6 DIUBAH JADI ARRAY 7:9 =
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .
. . . . .

Press any key to continue . . .
```

Gambar 21. Tampil hasil pengenalan karakter pola 4,5 dan 6

## V. KESIMPULAN DAN SARAN

Dari hasil percobaan pengenalan pola karakter menggunakan algoritma perceptron di atas diperoleh bahwa dengan *learning rate* = 1, *bias awal* = 0, *threshold* = 0.2 dapat mengenali pola huruf A dengan 3 epoch dimana target pola ( $y / f(\text{net})$ ) sama dengan target yang diinginkan ( $t$ ). Algoritma perceptron tersebut juga dapat digunakan untuk pengenalan pola karakter lain.

## DAFTAR PUSTAKA

- [1] Siang, Jong Jek. *Jaringan Saraf Tiruan dan Pemrogramannya Menggunakan Matlab*. Yogyakarta: 2005
- [2] Haykin, S., *Neural Networks, a Comprehensive Foundation*, Prentice Hall, 1994
- [3] Demuth, H., Beale, M., *Neural Network Toolbox, For Use with MATLAB*, The MathWorks, 2001
- [4] Stephens, Rod., *Visual Basic 2005 Programmer's Reference*, Wiley Publishing, Indianapolis: 2005

